

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Jürgen ANGELE

Appl. No.: Continuation-in-Part of
PCT/EP02/05347

Confirmation No.: Not Yet Assigned

Filed: November 13, 2003

For: COMPUTER SYSTEM

Art Unit: Not Yet Assigned

Examiner: Not Yet Assigned

Atty. Docket No.: 37934-191592

Customer No.

26694

PATENT TRADEMARK OFFICE

Submission of Certified Copy of Priority Document

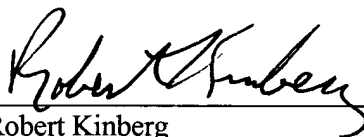
Commissioner for Patents
P.O. Box 1450
Alexandria, VA. 22313-1450

Sir:

Submitted herewith is a certified copy of Application No. 101 23 959.9 filed on
May 17, 2001 in Germany, the priority of which is claimed in the present application under the
provisions of 35 U.S.C. 119.

Date: 11/13/03

Respectfully submitted,



Robert Kinberg
Registration No. 26,924
VENABLE
P.O. Box 34385
Washington, D.C. 20043-9998

Telephone: (202) 344-4800

Telefax: (202) 344-8300

RK/cdw
::ODMA\PCDOCS\DC2DOCS\1498147\1



Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen: 101 23 959.9

Anmeldetag: 17. Mai 2001

Anmelder/Inhaber: Ontoprise GmbH, Karlsruhe/DE

Bezeichnung: Rechnersystem

IPC: G 06 F 17/30

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 15. September 2003
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

A handwritten signature in black ink, likely belonging to the official representing the German Patent and Trademark Office.

Wallner

P5040200

Ontoprise GmbH

76131 Karlsruhe, DE

5 Patentansprüche

1. Rechnersystem mit Mitteln zur Speicherung von Daten, mit Mitteln zur
Zuordnung der Daten zu Klassen wenigstens einer ein Objektmodell bil-
denden Klassenstruktur, mit Mitteln zur Speicherung von Regeln zur
10 Verknüpfung von Komponenten von Klassenstrukturen, mit einer Infe-
renzeinheit zur Generierung von Ausgangsgrößen durch Auswertung von
Regeln und mit einem Editor zur Generierung von Regeln und/oder Klas-
senstrukturen und/oder von Komponenten hiervon.
2. Rechnersystem nach Anspruch 1, dadurch gekennzeichnet, dass die
15 Komponenten von Klassenstrukturen von einzelnen Klassen oder von
Attributen, welche Klassen zugeordnet sind und welche innerhalb einer
Klassenstruktur weitervererbt sind, gebildet sind.
3. Rechnersystem nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass
bei einer Klassenstruktur nur Einfachvererbungen zugelassen sind.
- 20 4. Rechnersystem nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass
bei einer Klassenstruktur Mehrfachvererbungen zugelassen sind.
5. Rechnersystem nach einem der Ansprüche 2 – 4, dadurch gekennzeich-
net, dass mittels der Regeln Klassen und/oder Attribute und/oder Daten
wenigstens einer Klassenstruktur verknüpft sind.
- 25 6. Rechnersystem nach einem der Ansprüche 1 – 5, dadurch gekennzeich-
net, dass das oder jedes eine Klassenstruktur bildende Objektmodell als
DAML+OIL-Modell ausgebildet ist.

7. Rechnersystem nach Anspruch 6, dadurch gekennzeichnet, dass die Regeln in der Regelsprache DAML-L abgefasst sind.
8. Rechnersystem nach einem der Ansprüche 1 – 7, dadurch gekennzeichnet, dass in der Inferenzeinheit (5) zur Auswertung von Regeln Beziehungen zwischen Klassen und/oder Attributen und/oder Daten wenigstens einer Klassenstruktur gebildet werden.
9. Rechnersystem nach einem der Ansprüche 1 – 8, dadurch gekennzeichnet, dass die Ausgangsgrößen von Klassen und/oder Attributen und/oder Daten wenigstens einer Klassenstruktur gebildet sind.
10. Rechnersystem nach einem der Ansprüche 1 – 9, dadurch gekennzeichnet, dass die Ausgangsgrößen von neuen Beziehungen zwischen Daten und/oder Klassen und/oder Attributen wenigstens einer Klassenstruktur gebildet sind.
11. Rechnersystem nach einem der Ansprüche 1 – 10, dadurch gekennzeichnet, dass mittels des Editors und eines dem Editor zugeordneten Terminals (6) Klassen einer Klassenstruktur eingebbar sind.
12. Rechnersystem nach Anspruch 11, dadurch gekennzeichnet, dass mittels des Editors und des dem Editor zugeordneten Terminals (6) Attribute von Klassen eingebbar sind.
13. Rechnersystem nach Anspruch 12, dadurch gekennzeichnet, dass mittels des Editors den Attributen und/oder Klassen vorgegebene Wertebereiche zugeordnet werden.
14. Rechnersystem nach einem der Ansprüche 1 – 13, dadurch gekennzeichnet, dass mittels des Editors Regeln programmierbar sind.

15. Rechnersystem nach einem der Ansprüche 1 – 13, dadurch gekennzeichnet, dass in den Mitteln zur Speicherung von Regeln eine Anzahl von Regeln abgespeichert ist, und dass mittels des Editors eine vorgegebene Anzahl von Regeln auswählbar ist.
- 5 16. Rechnersystem nach Anspruch 15, dadurch gekennzeichnet, dass im Editor eine vorgegebene Anzahl definiert ist, wobei jedem Axiom ein bestimmter Regeltyp zugeordnet ist, und wobei durch Auswahl eines Axioms im Editor eine Regel dieses Regeltyps generiert wird.
- 10 17. Rechnersystem nach Anspruch 16, dadurch gekennzeichnet, dass die Axiome von die Funktionen der Regeln definierenden mathematischen Begriffen gebildet sind.
18. Rechnersystem nach einem der Ansprüche 1 – 17, dadurch gekennzeichnet, dass der Editor eine Graphik-Oberfläche aufweist.
- 15 19. Rechnersystem nach Anspruch 18, dadurch gekennzeichnet, dass mittels der Graphik-Oberfläche des Editors Regeln erstellbar sind.
- 20 20. Rechnersystem nach Anspruch 19, dadurch gekennzeichnet, dass in dem Editor Klassen und/oder Attribute und/oder Daten sowie Beziehungen zwischen diesen als Graphik-Komponenten im Editor visualisiert sind, und dass durch Verbinden von Graphik-Komponenten Regeln definierbar sind.
21. Rechnersystem nach Anspruch 20, dadurch gekennzeichnet, dass mittels des Editors eine Eingabe von Daten erfolgt.
- 25 22. Rechnersystem nach einem der Ansprüche 1 – 21, dadurch gekennzeichnet, dass mittels des Editors eine Überprüfung von Regeln und Klassenstrukturen durchführbar ist.

23. Rechnersystem nach Anspruch 22, dadurch gekennzeichnet, dass bei der Generierung fehlerhafter Regeln oder Klassenstrukturen im Editor eine Fehlermeldung generiert wird.
- 5 24. Rechnersystem nach einem der Ansprüche 1 – 23, dadurch gekennzeichnet, dass dieses wenigstens eine Ein-/Ausgabeeinheit aufweist.
25. Rechnersystem nach Anspruch 24, dadurch gekennzeichnet, dass über die Ein-/Ausgabeeinheit Abfragebefehle zur Aktivierung der Inferenzeinheit (5) eingebbar sind.
- 10 26. Rechnersystem nach einem der Ansprüche 24 oder 25, dadurch gekennzeichnet, dass über die Ein-/Ausgabeeinheit Daten eingebbar und/oder Ausgangsgrößen ausgebbbar sind.

P5040200

Ontoprise GmbH

76131 Karlsruhe, DE

5 Rechnersystem

Die Erfindung betrifft ein Rechnersystem.

Derartige Rechnersysteme können von vernetzten Rechneinheiten gebildet sein, auf welchen als Mittel zur Speicherung von Daten insbesondere Datenbanksysteme integriert sind. Insbesondere können die Rechneinheiten auch
10 an das Internet angeschlossen sein, so dass die Datenbanksysteme über das Internet abfragbar sind.

Generell enthalten derartige Datenbanksysteme große Datensätze, die mit vorgegebenen Abfragebefehlen abfragbar sind. Ein wesentliches Problem besteht insbesondere bei Datenbanksystemen, in welchen große Datenmengen gespeichert sind, darin, geeignete Abfragebefehle zu definieren, um so zu den ge-
15 wünschten Rechercheergebnisse zu gelangen.

Besonders schwierig gestalten sich derartige Recherchen dann, wenn zu allgemeinen Themenstellungen Daten benötigt werden, jedoch über diese Themenstellungen nur wenige recherchierbare Daten bekannt sind.

20 Ein Beispiel hierfür kann folgende Aufgabenstellung sein. Ein Rechnersystem umfasst Datenbanksysteme, in welchem technische, medizinische und kaufmännische Veröffentlichungen unterschiedlicher Art abgespeichert sind. Ein Benutzer dieses Datenbanksystems kennt den Namen eines Autors einer Veröffentlichung. Dabei weiß er nur, dass es sich hierbei um eine technische Veröffentlichung handelt. Der einzige recherchierbare Begriff, der dem Benutzer zur
25 Verfügung steht, ist der Name des Autors der Veröffentlichung. Dieser Name bildet die Eingangsgröße für die Recherche, die über eine Abfrageeinheit in das

Rechnersystem eingegeben wird. Da weitere Informationen nicht vorliegen, muss der Benutzer sämtliche Abfrageergebnisse zu dem recherchierten Namen durchsehen, um dort anhand gegebenenfalls vorliegenden zusätzlich ermittelten Informationen zu dem Autor der gewünschten Veröffentlichung zu gelangen.

5 Eine derartige zusätzliche manuelle Auswertung ist äußerst umständlich und beinhaltet zudem erhebliche Fehlerquellen, so dass das Recherchenergebnis mit einer erheblichen Ungenauigkeit behaftet ist.

Der Erfindung liegt die Aufgabe zugrunde ein Rechnersystem der eingangs genannten Art so auszubilden, dass ein möglichst umfassender, einfacher und flexibler Zugriff auf Informationen, die im Rechnersystem gespeichert sind, gewährleistet ist.

10

Zur Lösung dieser Aufgabe sind die Merkmale des Anspruchs 1 vorgesehen. Vorteilhafte Ausführungsformen und zweckmäßige Weiterbildungen der Erfindung sind in den Unteransprüchen beschrieben.

15 Das erfindungsgemäße Rechnersystem weist Mittel zur Speicherung von Daten, Mittel zur Zuordnung der Daten zu Klassen wenigstens einer ein Objektmodell bildenden Klassenstruktur und Mittel zur Speicherung von Regeln zur Verknüpfung von Komponenten von Klassenstrukturen auf. Ebenso weist das erfindungsgemäße Rechnersystem eine Inferenzeinheit zur Generierung von

20 Ausgangsgrößen durch Auswertung von Regeln und einen Editor zur Generierung von Regeln und/oder Klassenstrukturen und/oder von Komponenten hiervon auf.

Der Grundgedanke der Erfindung besteht somit darin, dass die im Rechnersystem abgespeicherten Daten innerhalb wenigstens eines Objektmodells, vorzugsweise innerhalb mehrerer Objektmodelle, strukturiert sind. Derartige Ontologien bildende Objektmodelle stellen Klassenstrukturen dar, die hierarchisch

25 oder in azyklischen Graphen strukturierte Klassen aufweisen, wobei den Klassen vorzugsweise mehrere Attribute zugeordnet sind, die innerhalb einer Klas-

senstruktur weitervererbt sind. Erfindungsgemäß erfolgt der Zugriff auf im Rechnersystem abgespeicherte Informationen nicht oder nicht allein durch Abfrage von dort abgespeicherten Daten.

5 Anstelle dessen weist das erfindungsgemäße Rechnersystem eine vorgegebene Anzahl von Regeln auf, die wenigstens einer Inferenzeinheit zugeordnet sind.

Mittels der Regeln können Attribute wenigstens einer Klassenstruktur und/oder Klassen wenigstens einer Klassenstruktur sowie gegebenenfalls auch abgespeicherte Daten miteinander verknüpft werden. Dabei stellen die Regeln die logischen Verknüpfungsvorschriften dar, die die einzelnen vorgenannten Elemente
10 in vorgegebener Weise miteinander in Beziehung setzen. In der Inferenzeinheit wird eine Auswertung vorgenommen, in dem diesen Regeln konkrete Werte für die Attribute, Klassen und/oder Daten zugeordnet werden, wodurch bestimmte Ausgangsgrößen generiert werden.

Mittels des erfindungsgemäßen Editors können vom Anwender des Rechnersystems die Regeln und/oder Klassenstrukturen selbst generiert werden. Hierzu
15 weist der Editor entsprechende Programmieroberflächen oder Graphik-Oberflächen auf, mittels derer die Eingabe der entsprechenden Komponenten der Regeln und/oder Klassenstrukturen erfolgen kann. Mittels des Editors sind somit die Strukturen der Regeln sowie die Klassenstrukturen auf einfache Weise flexibel vorgebbar und veränderbar.
20

Insbesondere können zur Generierung der Regeln Graphik-Oberflächen vorgesehen sein, mittels derer Regeln vollständig definiert werden können. Alternativ können Regeln im Editor frei programmiert werden, das heißt der Anwender gibt die Regeln durch Eingabe von Programmierbefehlen vollständig vor.
25 In einer weiteren vorteilhaften Ausführungsform der Erfindung sind im Editor vorgegebene Axiome auswählbar. Durch Auswahl eines Axioms durch den Anwender wird eine diesem Axiom eindeutig zugeordnete Regel, die vorzugs-

weise im Editor abgespeichert ist, aktiviert. Auf diese Weise kann der Anwender in besonders einfacher Weise die gewünschten Regeln generieren.

Ein wesentlicher Vorteil des erfindungsgemäßen Rechnersystems besteht darin, dass die Abfrage und Auswertung von Informationen nicht auf die Ebene der in dem Rechnersystem abgespeicherten Daten beschränkt ist. Vielmehr sind die
5 Abfragen auf die Strukturelemente der Objektmodelle, in welchen die Daten strukturiert sind, erweitert. Damit können bereits mit rudimentären und einfachen Anfragen beziehungsweise Eingabewerte auch komplexe Sachverhalte und Zusammenhänge aus den im Rechnersystem abgespeicherten Informationen extrahiert werden.
10

Ein einfaches Beispiel für das erfindungsgemäße Rechnersystem ist eine Rechereinheit, auf welchem ein Datenbanksystem integriert ist. Durch Abfragen bestimmter Klassen oder Attribute werden als Ausgangsgrößen Untermengen von Daten erhalten, ohne dass die Daten selbst unmittelbar abgefragt werden
15 müssen. Derartige Abfrageschemas sind besonders deshalb vorteilhaft, da mit den Klassen und Attributen von Klassenstrukturen Klassifikationen von Daten nach bestimmten Kriterien und Eigenschaften vorgenommen werden können, die auf einfache Weise recherchiert werden können. Beispielsweise können Personaldaten Klassenstrukturen aufweisen, die in verschiedene Klassen entsprechend der Hierarchie von Mitarbeitern in einer Firma untergliedert sind.
20 Eine derartige Klassenstruktur kann in einer ersten Ebene die Klasse „Angestellte“ enthalten, die in Unterklassen „technische Angestellte“ und „kaufmännische Angestellte“ verzweigt. Diese Unterklassen können in weitere Unterklassen verzweigt sein. Diesen Klassen können als Attribute das Geschlecht
25 des Mitarbeiters oder andere Eigenschaften zugeordnet sein.

Mit dem erfindungsgemäßen Rechnersystem können durch Abfragen bestimmte Klassen und Attribute die diesen Elementen zugeordneten Mitarbeitern ermittelt werden, ohne dass deren konkrete Daten wie zum Beispiel Name, Adresse und Abteilungsbezeichnung innerhalb der Firma konkret abgefragt

werden müssen. Durch die Abfragemöglichkeit in der oberhalb der Datenebene liegenden Klassen- und/oder Attributebene entsteht ein besonders mächtiges und flexibles Abfragesystem, welches die Recherchemöglichkeiten im Vergleich zu herkömmlichen Datenbanksystemen beträchtlich erweitert.

- 5 Weiterhin können durch Auswertung der Regeln Beziehungen zwischen Daten, Attributen und/oder Klassen abgeleitet werden, die in dieser Form nicht im Datenbanksystem abgespeichert sind. Somit kann mit dem erfindungsgemäßen Rechnersystem aus bekannten, abgespeicherten Größen neues Wissen abgeleitet werden.
- 10 Ein weiterer Vorteil des erfindungsgemäßen Rechnersystems besteht darin, dass ein Benutzer mehrere Abfragebegriffe als Eingangsgrößen in das Rechnersystem eingeben kann, ohne selbst eine Unterscheidung treffen zu müssen, ob es sich bei diesen Abfragebegriffen um zu recherchierende Daten, Klassen oder Attribute handelt. Diese Abfragebegriffe werden als Eingangsgrößen in die
- 15 Inferenzeinheit mit den dem jeweiligen Abfragebefehl zugeordneten Regeln verknüpft. Anhand der Regeln erfolgt die Zuordnung der Abfragebegriffe zu den Daten, Klassen und/oder Attributen eines Objektmodells. Als Ausgangsgrößen werden Untermengen von Daten erhalten, die entsprechend der Ausbildung der Regeln in einem vorgegebenen Verhältnis zueinander stehen. Im einfachsten Fall erfolgt eine Verknüpfung der Abfragebegriffe zu einer einzelnen
- 20 Ausgangsgröße.

Die Erfindung wird im nachstehenden anhand der Zeichnungen erläutert. Es zeigen:

25 Figur 1: Aufbau eines Ausführungsbeispiels des erfindungsgemäßen Rechnersystems.

Figur 2: Objektmodelle zur Strukturierung der auf dem Rechnersystem gemäß Figur 1 gespeicherten Daten.

Figur 3: Ausführungsbeispiel einer Graphik-Oberfläche des Editors für das Rechnersystem gemäß Figur 1.

Figur 1 zeigt schematisch den prinzipiellen Aufbau eines ersten Ausführungsbeispiels des erfindungsgemäßen Rechnersystems 1.

5 Das Rechnersystem 1 umfasst mehrere Rechneinheiten 2, welche über ein Rechnerleitungen 3 aufweisendes Netzwerk miteinander verbunden sind. Eine der Rechneinheiten 2 bildet einen Zentralrechner, auf welchem Daten abgespeichert sind. Die Mittel zur Speicherung der Daten sind von einem Datenbanksystem 4 gebildet. Zur Durchführung und Auswertung von Abfragen in
10 dem Datenbanksystem 4 ist als Abfrageeinheit eine Inferenzeinheit 5 vorgesehen. Zudem weist die Rechneinheit 2 als Ein-/Ausgabeeinheit ein Terminal 6 auf, über welches ein Editor bedienbar ist.

Mehrere Benutzer können über weitere an das Netzwerk angeschlossene Rechneinheiten 2, wie zum Beispiel Personalcomputer, Zugang zum Datenbanksystem 4 erhalten. Hierzu weisen die Rechneinheiten 2 geeignete Ein-/Ausgabeeinheiten, vorzugsweise in Form von Terminals 6 auf.
15

Insbesondere kann das Netzwerk vom Internet gebildet sein. In diesem Fall weisen die Rechneinheiten 2 entsprechende Internetanschlüsse auf.

Zur Strukturierung der im Datenbanksystem 4 abgespeicherten Daten werden
20 Objektmodelle, sogenannte Ontologien, eingesetzt. Ein Objektmodell weist eine Struktur von Klassen auf, wobei die Struktur als hierarchische Struktur ausgebildet sein kann. Bei hierarchischen Strukturen sind Klassen einer vorgegebenen Ebene jeweils genau einer Klasse einer darüber liegenden Ebene zugeordnet, das heißt es sind nur Einfachvererbungen zugelassen. Allgemein
25 kann die Klassenstruktur auch als azyklischer Graph ausgebildet sein, bei welchem Mehrfachvererbungen zugelassen sind.

Figur 2 zeigt beispielhaft zwei derartige, hierarchische Klassenstrukturen, die jeweils ein Objektmodell bilden. Das erste Objektmodell enthält eine Klasse „Veröffentlichungen“, welcher als Unterklasse „Vorträge“ und „Dokumente“ zugeordnet sind. Das zweite Objektmodell enthält eine Klasse „Personen“,
5 welcher als Unterklassen „Selbständige“ und „Angestellte“ zugeordnet sind. Der Unterklasse „Angestellte“ sind als weitere Unterklassen „technische Angestellte“ und „kaufmännische Angestellte“ zugeordnet.

Den Klassen jeweils einer hierarchischen Klassenstruktur sind bestimmte Attribute zugeordnet. Dabei wird ein Attribut, welches einer Klasse wie zum Beispiel der Klasse „Personen“ zugeordnet ist, an die dieser Klasse untergeordneten Unterklassen weiter vererbt. Ein derartiges Attribut kann beispielsweise ein Name sein. Diese Attribut wird innerhalb der Klassenstruktur, im vorliegenden
10 Beispiel an die untergeordneten Klassen „Selbständige“ und „Angestellte“ sowie auch die dieser Klasse zugeordneten Unterklassen „kaufmännische Angestellte“ und „technische Angestellte“ vererbt. Auf diese Weise entsteht eine besonders effiziente Strukturierung der Daten in dem Datenbanksystem 4.

Zur Durchführung von Abfragen in dem Datenbanksystem 4 sind der Inferenzeinheit 5 Regeln zugeordnet. Diese Regeln sind in der Inferenzeinheit 5 selbst oder in einer nicht dargestellten, der Inferenzeinheit 5 zugeordneten Speichereinheit abgespeichert.
15

Die Objektmodelle sowie die Sprache, in der die Regeln abgefasst sind, können unterschiedliche Ausprägungen aufweisen. Vorzugsweise werden Objektmodelle des Typs DAML+OIL und als Regelsprache DAM-L verwendet.

Zur Durchführung von Abfragen in dem Datenbanksystem 4 werden in eine
25 Ein-/Ausgabeeinheit definierte Abfragebefehle eingegeben. Je nach Ausbildung des Abfragebefehls wird in der Inferenzeinheit 5 eine Folge von Regeln abgearbeitet. Da es sich bei den Regeln um ein deklaratives System handelt, spielt die Reihenfolge der Definition der Regeln hierbei keine Rolle.

Die Regeln beinhalten Beziehungen in Form von logischen Verknüpfungen zwischen Klassen und/oder Attributen und/oder Daten des Datenbanksystems

4. In der Inferenzeinheit 5 werden die einem Abfragebefehl zugeordneten Regeln zur Generierung definierter Ausgangsgrößen ausgewertet. Zweckmäßigerweise werden die Ausgangsgrößen dann über die Ein-/Ausgabeeinheit ausgegeben.

Durch die Verknüpfung von Attributen und Klassen über eine vorgegebene Anzahl von Regeln können auf einfache Weise Untermengen von Daten im Datenbanksystem 4 abgefragt werden, ohne dass dabei in den Abfragebefehlen auf bestimmte Daten Bezug genommen werden muss.

Mit der Abfragemöglichkeit auf Klassen- und Attributebene wird gegenüber herkömmlichen Datenbanksystemen 4, bei welchen die Abfragebefehle auf die Datenebene begrenzt sind, eine erhebliche Erweiterung und Flexibilisierung der Abfragemöglichkeiten erreicht.

Ein derartiger Abfragebefehl kann beispielsweise wie folgt ausgebildet sein:

„Gebe die Namen von allen gespeicherten Daten aus, die in der Hierarchie der Klassenstruktur des Objektmodells „Personen“ unterhalb der Ebene „Angestellte“ liegen“.

Als Ausgangsgröße werden in diesem Fall dem Benutzer die Namen aller im Datenbanksystem 4 gespeicherter technischer und kaufmännischer Angestellten angezeigt.

In einer weiteren vorteilhaften Ausführungsform können mit den einzelnen Abfragebefehlen zugeordneten Regeln Beziehungen zwischen unterschiedlichen Attributen, Klassen und/oder Daten hergestellt werden. Dabei können mit den Regeln insbesondere auch Attribute, Klassen und/oder Daten verschiedener Klassenstrukturen miteinander verknüpft werden.

Dabei ist besonders vorteilhaft, dass der Benutzer in einem Abfragebefehl lediglich die Begriffe, nach welchen die Suche durchgeführt wird, vorzugsweise in einer Folge hintereinander eingeben muss. Dabei braucht der Benutzer keine Definitionen vorzunehmen, ob es sich bei diesen Begriffen um Klassen, Attribute oder Daten handelt. Zudem muss der Benutzer hierbei keinerlei Eingriff in die Struktur der Regeln vornehmen, die einem bestimmten Abfragebefehl zugeordnet sind. Die Zuordnung der Begriffe zu den Regeln und die Abarbeitung der Regeln erfolgt selbsttätig in der Inferenzeinheit 5.

Ein Beispiel für eine derartige Abfrage kann wie folgt ausgebildet sein. Ein Benutzer möchte wissen, über welche Kenntnisse eine ihm bekannte Persönlichkeit mit Namen Mustermann verfügt.

Als Abfragebefehl gibt der Benutzer die beiden Suchgrößen „Mustermann“ und „Kenntnisse“ in die Ein-/Ausgabeeinheit ein.

In der Inferenzeinheit 5 erfolgt eine Auswertung von Regeln, die diesem Abfragebefehl zugeordnet sind. Eine derartige Regel kann beispielsweise lauten:

„Wenn eine Person ein Dokument schreibt und das Dokument von einem Thema handelt, dann hat die Person Kenntnisse zu diesem Thema“.

Diese Regel verknüpft die Klassen „Person“ und „Dokument“ zweier verschiedener Klassenstrukturen. Dabei ist Bezug auf das Thema von Dokumenten genommen, wobei beispielsweise die Themen von Dokumenten als Daten der Klasse „Dokument“ zugeordnet sind.

Als Ausgangsgröße dieser Regel wird erhalten, ob eine Person „Kenntnisse“ zu diesem Thema hat.

Wie aus diesem Beispiel ersichtlich, werden durch derartige Verknüpfungen nicht nur im Datenbanksystem 4 abgespeicherte Informationen abgefragt.

Vielmehr werden mit derartigen Regeln Beziehungen zwischen Elementen der Datenbanksysteme 4 hergestellt, um daraus gegebenenfalls neue Kenngrößen abzuleiten. Dies bedeutet, dass mittels der Regeln aus den im Datenbanksystem 4 abgespeicherten Daten neues Wissen ableitbar ist, welches in dieser Form in dem Datenbanksystem 4 nicht abgespeichert ist.

Die Auswertung der oben genannten Regel in Abhängigkeit der eingegebenen Eingangsgrößen „Kenntnisse“ und „Mustermann“ erfolgt in der Inferenzeinheit 5 anhand einem dort abgespeicherten Zuordnungsschema, welches im vorliegenden Fall wie folgt ausgebildet ist:

10 Mustermann ist eine Person.

Mustermann ist Autor einer Diplomarbeit.

Die Diplomarbeit hat zum Thema Biotechnologie.

Die Diplomarbeit ist ein Dokument.

15 Unter Auswertung dieser Zuordnungen auf die genannte Regel ergibt sich als Ergebnis, dass Mustermann Kenntnisse über Biotechnologie hat. Dieses Ergebnis wird vorzugsweise über die Ein-/Ausgabeeinheit ausgegeben.

Dabei besteht ein wesentlicher Unterschied zu bekannten Datenbanksystemen 4 darin, dass das Rechercheergebnis „Mustermann hat Kenntnisse über Biotechnologie“ weder durch eine Datenbankabfrage des Begriffs „Kenntnisse“ noch des Begriffs „Biotechnologie“ erhalten wurde.

20 Eine Abfrage des Begriffs „Biotechnologie“ in einem herkömmlichen Datenbanksystem 4 würde voraussetzen, dass der Benutzer bereits detaillierte Vorkenntnisse über das Wissen von Mustermann hat. Zudem müsste in einem der

Person Mustermann zugeordneten Datensatz explizit der Begriff „Biotechnologie“ hinterlegt sein.

5 Eine Abfrage des Begriffs „Kenntnisse“ wäre in einem herkömmlichen Datenbanksystem 4 prinzipiell nicht sinnvoll, da dadurch keine Zuordnung des abstrakten Begriffs „Kenntnisse“ zu einem konkreten Faktum „Biotechnologie“ erfolgen kann.

10 Demgegenüber werden bei dem erfindungsgemäßen Rechnersystem 1 abstrakte Begriffe wie Klassen und/oder Attribute über Regeln verknüpft, welche wie im vorliegenden Fall als Ausgangsgrößen neue Kenngrößen liefern. Diese können wiederum abstrakte Größen bilden, die unmittelbar vom Benutzer recherchiert werden können. Die Zuordnung konkreter Werte zu den abstrakten Größen des Regelwerks erfolgt dann selbsttätig in der Inferenzeinheit 5.

15 Wie aus diesem Beispiel ersichtlich bedarf es bei dem erfindungsgemäßen Rechnersystem 1 im Vergleich zu herkömmlichen Datenbanksystemen 4 erheblich weniger Vorkenntnisse und damit auch weniger Eingaben um zu exakten Rechercheergebnissen zu gelangen.

Die Objektmodelle sowie die der Inferenzeinheit 5 zugeordneten Regeln sind zumindest teilweise über den Editor vorgebbar, wobei der Editor über das dem Zentralrechner zugeordnete Terminal 6 von einem Benutzer bedienbar ist.

20 Der Editor weist Programmier-Oberflächen und/oder Graphik-Oberflächen auf, über welche von Regeln und/oder Klassenstrukturen komponentenweise oder komplett vom Benutzer generiert werden können.

25 Der Aufbau einer Klassenstruktur kann beispielsweise graphisch generiert werden. Die Klassenstruktur wird dann mittels graphischer Elemente erzeugt. Ein Beispiel für eine derartige graphische Darstellung von Klassenstrukturen ist in Figur 2 dargestellt.

Alternativ kann die Klassenstruktur durch Einprogrammieren der entsprechenden Komponenten erfolgen.

Für die Objektmodelle gemäß Figur 2 bedeutet dies beispielsweise, dass für die einzelnen Klassen entsprechend hierarchisch gegliederte Verzeichnisse im Editor angelegt werden.

Eine den Klassenstrukturen gemäß Figur 2 entsprechende Verzeichnisstruktur ist in der nachstehenden Tabelle aufgeführt.

Tabelle 1:

- Personen
 - Selbständige
 - Angestellte
 - * kaufmännische Angestellte
 - * technische Angestellte
- Veröffentlichungen
 - Vorträge
 - Dokumente

Hierbei ist das der Klasse „Personen“ entsprechende Hauptverzeichnis in Unterverzeichnisse „Selbständige“ und „Angestellte“ untergliedert. Entsprechend der Zuordnung der Unterklassen „kaufmännische Angestellte“ und „technische Angestellte“ zur übergeordneten Klasse „Angestellte“ sind dem Verzeichnis „Angestellte“ in Tabelle 1 zwei entsprechende Unterverzeichnisse zugeordnet.

Den Klassen einer auf dieser Weise generierten Klassenstruktur können anschließend Attribute zugeordnet werden.

Die Programmier-Oberfläche ist dabei derart ausgebildet, dass bei Anklicken einer Klasse einer Klassenstruktur ein Fenster geöffnet wird, in welches die
5 einzelnen Attribute als Variable eingetragen werden. Wird beispielsweise vom Benutzer die Klasse „Person“ durch Anklicken des entsprechenden Verzeichnisses angeklickt, so können in das entsprechende Fenster Attribute wie zum Beispiel Vornamen, Zunamen, Geburtstag und dergleichen eingegeben werden. Zudem werden den einzelnen Attributen vorgegebene Wertebereiche zugeordnet.
10

Auf diese Weise sind vom Benutzer komplette Klassenstrukturen vorgebbbar, welchen anschließend die einzelnen Daten zugeordnet werden. Die Dateneingabe kann über den Editor erfolgen. Weiterhin können zur Dateneingabe dezentrale Ein-/Ausgabeeinheiten vorgesehen sein.

15 Mittels des Editors können vom Benutzer des Weiteren Regeln generiert werden, die zur Generierung der Ausgangsgrößen in der Inferenzeinheit 5 ausgewertet werden.

Eine erste Möglichkeit zur Generierung der Regeln besteht darin, dass der Benutzer die Regeln frei programmiert, wobei die Regeln in der Programmiersprache DAML-L abgefasst werden. Hierzu ist es notwendig, dass der Benutzer
20 die Programmiersprache beherrscht.

Eine einfachere Möglichkeit zur Generierung von Regeln besteht darin, dass im Editor eine vorgegebene Anzahl von Axiomen definiert ist, welche dem Benutzer in einem Fenster der Programmier-Oberfläche des Editors angezeigt werden.
25 Jedem dieser Axiome ist dabei ein definierter Regeltyp zugeordnet.

Der Benutzer wählt ein bestimmtes Axiom durch Anklicken in der Programmier-Oberfläche aus, wodurch eine Regel des entsprechenden Regeltyps generiert wird.

- 5 Die Axiome sind dabei vorzugsweise von definierten mathematischen Begriffen gebildet.

Beispiele für derartige mathematisch definierte Axiome sind binäre Beziehungen wie symmetrische, antisymmetrische, asymmetrische, inverse, reflexive, irreflexive oder transitive Beziehungen. Weiterhin können Axiome die Disjunktheit von Klassen ausdrücken.

- 10 Schließlich besteht auch die Möglichkeit für den Benutzer Regeln graphisch zu generieren. Eine hierfür geeignete Graphik-Oberfläche ist in Figur 3 dargestellt.

Figur 3 zeigt einen Zustand der Graphik-Oberfläche, bei welchem drei verschiedene Fenster geöffnet sind, wobei jedes Fenster eine bestimmte Klasse enthält.

- 15 Zwei der Fenster sind identisch und beziehen sich auf die Klasse „Person“. Ein drittes Fenster bezieht sich auf die Klasse „Dokument“. Die Klassen sind Bestandteil der in Figur 2 dargestellten Klassenstrukturen.

- 20 Der Klasse „Person“ sind gemäß dem in Figur 3 dargestellten Beispiel als Attribute Personaldaten wie Name, Email-Adresse, Telefonnummer und Adresse zugeordnet. Weiterhin ist der Klasse „Person“ als Attribut die Beziehung „hat Ahnung von“ oder gleichbedeutend „hat Kenntnisse über“ zugeordnet.

Weiterhin sind der Klasse „Person“ folgende Beziehungen (Relationen) zugeordnet:

- Projekt

- ist Autor von
- leitet Projekt

Der Klasse „Dokument“ sind als Attribute Kenngrößen von Dokumenten, nämlich der Titel und das Thema eines Dokuments zugeordnet. Weiterhin sind
5 der Klasse „Dokument“ folgende Beziehungen zugeordnet:

- hat Datum
- hat Autor
- hat Zweck

Zur graphischen Generierung einer Regel werden wie in Figur 3 dargestellt
10 mehrere Komponenten der in Figur 3 dargestellten Klassen durch Linien miteinander verbunden. Im vorliegenden Beispiel wird über eine erste Linie die Beziehung „ist Autor von“ in der Klasse „Person“ mit der Klasse „Dokument“ verbunden. Über eine zweite Linie ist das Attribut „über Thema“ der Klasse „Dokument“ mit dem Attribut „hat Ahnung von“ der Klasse „Person“ verbunden.
15 den.

Dabei sind die graphischen Elemente zur Generierung der Regel so ausgebildet, dass an den Endpunkten der ersten Linie und an einem Endpunkt der zweiten Linie Kreisflächen vorgesehen sind. Diese Kreisflächen zeigen auf Attribute, Klassen oder Beziehungen, welche die Bedingungen einer Regel definieren.
20 Am zweiten Endpunkt der zweiten Linie befindet sich ein Pfeil. Das Attribut „hat Ahnung von“ der Klasse „Person“, auf welche der Pfeil zeigt, bildet die Schlußfolgerung der Regel.

Somit wird durch die Vorgabe der beiden graphischen Elemente gemäß Figur 3, nämlich zwei durch Kreisflächen beziehungsweise einen Pfeil begrenzte Linien, folgende Regel generiert.

- 5 „Wenn eine Person Autor eines Dokuments über ein Thema ist, dann hat die Person Ahnung von diesem Thema“.

- 10 Schließlich wird mittels des Editors eine Überprüfung durchgeführt, ob die vom Benutzer eingegebenen Komponenten von Klassenstrukturen und/oder Regeln korrekt sind. Insbesondere erfolgt eine Konsistenz- und Vollständigkeitsprüfung. Für den Fall einer fehlerhaften Benutzereingabe wird vorteilhaftweise im Editor eine Fehlermeldung generiert.

P5040200

Ontoprise GmbH

76131 Karlsruhe, DE

5 Zusammenfassung

Die Erfindung betrifft ein Rechnersystem (1) mit Mitteln zur Speicherung von Daten, mit Mitteln zur Zuordnung der Daten zu Klassen wenigstens einer ein Objektmodell bildenden Klassenstruktur, mit Mitteln zur Speicherung von Regeln zur Verknüpfung von Komponenten von Klassenstrukturen, mit einer Inferenzeinheit (5) zur Generierung von Ausgangsgrößen durch Auswertung von Regeln und mit einem Editor zur Generierung von Regeln und/oder Klassenstrukturen und/oder von Komponenten hiervon.

15 Figur 1

P5040200

Ontoprise GmbH

76131 Karlsruhe, DE

5 Bezugszeichenliste

(1) Rechnersystem

(2) Rechnereinheiten

(3) Rechnerleitungen

10 (4) Datenbanksysteme

(5) Inferenzeinheit

(6) Terminal

Fig. 1

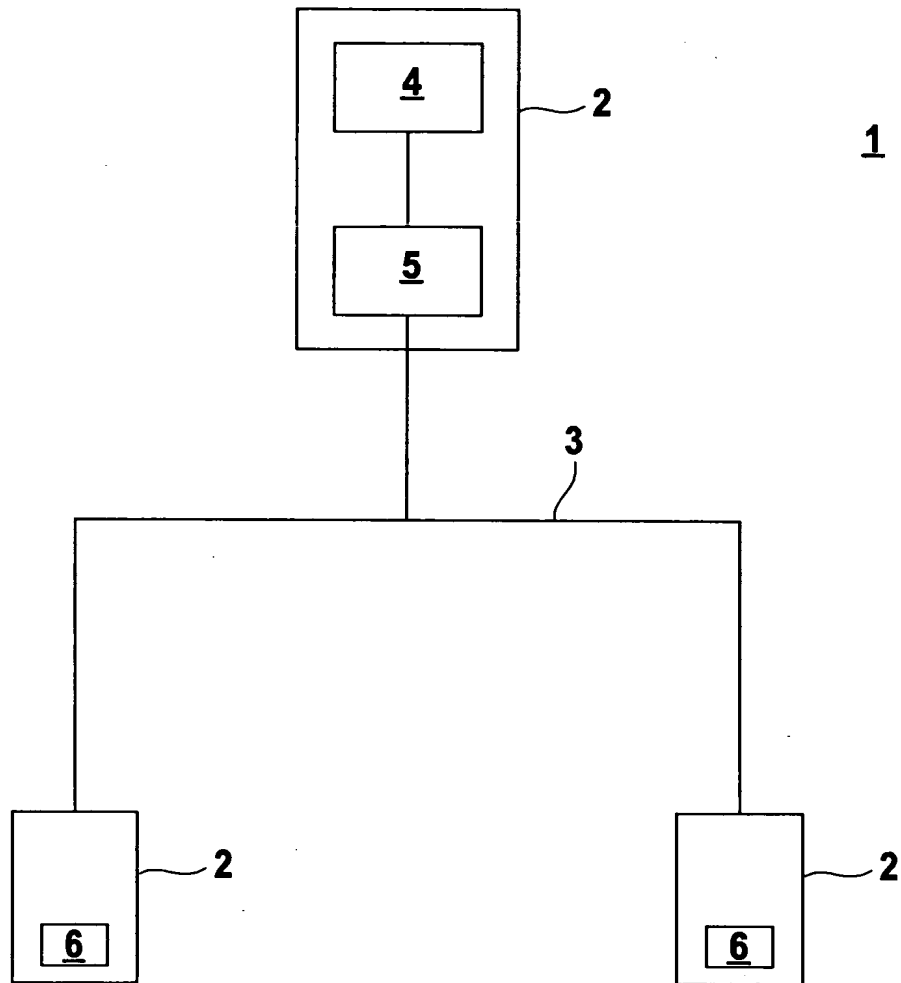
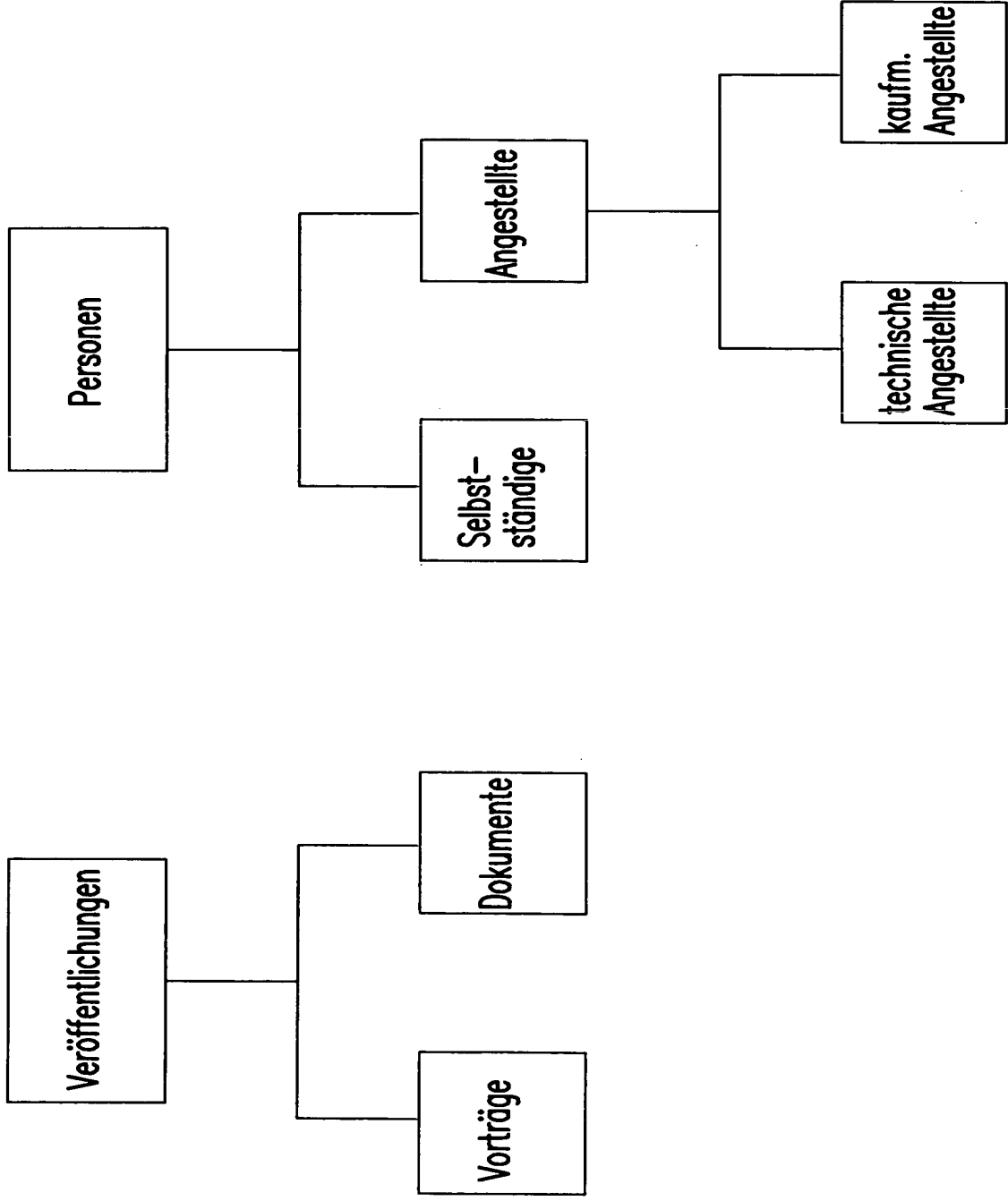


Fig. 2



Konzept - Microsoft Internet Explorer

Person

Id: Ob2

Attribut

name

email

telefon

adresse

hat_Ahnung_von

Relation

Id

Restriktion

projekt

Ist_Autor_von

leitet_Projekt

Projekt

Dokument

Projekt

Fertig

Lokales Intranet

Konzept - Microsoft Internet Explorer

Dokument

Id:

Attribut

titel

ueber_Thema

Relation

hat_Datum

hat_Autor

hat_Zweck

Relation Wert

Id

Restriktion

Zeitpunkt

Person

Zweck

Fertig

Lokales Intranet

Konzept - Microsoft Internet Explorer

Person

Id: Ob2

Attribut

name

email

telefon

adresse

hat_Ahnung_von

Relation

Id

Restriktion

projekt

Ist_Autor_von

leitet_Projekt

Projekt

Dokument

Projekt

Fertig

Lokales Intranet

Fig. 3